# SPYWOLF

## Security Audit Report

Completed on

**Sept 26, 2023**

# OVERVIEW

This audit has been prepared for **Fishing Tuna** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

*The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal*

- SPYWOLF Team -

# TABLE OF CONTENTS

# Fishing Tuna

## PROJECT DESCRIPTION

**According to their whitepaper:**

You don't have to be a professional! Cast your line and watch the float, catch fish and earn money with pleasure!

Daily profit with a starting set of 1%. Buy and hold TUNA tokens to improve your equipment and get increased daily profit up to 1.6%.

**Release Date:** Launched Sept 21st, 2023

**Category:** Staking

# CONTRACT INFO

Token Name
FishingTunaContract

Symbol
N/A

Contract Address
0x4496e50fb325DCfdD15544e543dA6810f9D4420b

Network
Binance Smart Chain

Language
Solidity

Deployment Date
Sept 19, 2023

Contract Type
Staking

Total Supply
N/A

Status
Launched

# TAXES

Buy Tax
**4%**

Sell Tax
**none**

*Taxes cannot be changed

# Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

**Blockchain security tools used:**

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

# TOKEN TRANSFERS STATS

| | |
|---|---|
| **Transfer Count** | 1 |
| **Uniq Senders** | 1 |
| **Uniq Receivers** | 1 |
| **Total Amount** | 99999.99999999999 FLY |
| **Median Transfer Amount** | 99999.99999999999 FLY |
| **Average Transfer Amount** | 99999.99999999999 FLY |
| **First transfer date** | 2022-06-02 |
| **Last transfer date** | 2022-06-02 |
| **Days token transferred** | 1 |

# SMART CONTRACT STATS

| | |
|---|---|
| **Calls Count** | 641 |
| **External calls** | 641 |
| **Internal calls** | 0 |
| **Transactions count** | 641 |
| **Uniq Callers** | 241 |
| **Days contract called** | 8 |
| **Last transaction time** | 2023-09-26 08:52:39 UTC |
| **Created** | 2023-09-19 12:08:03 UTC |
| **Create TX** | 0xb8c4358de551670242d2f6eba045bd0fecb003d831fddc291c15fc23e81b09c3 |
| **Creator** | 0x73e3e1b0523d02279672843 1a2f1e4a13671a841 |

# VULNERABILITY CHECK

| | |
|---|---|
| Design Logic | Passed |
| Compiler warnings. | Passed |
| Private user data leaks | Passed |
| Timestamp dependence | Passed |
| Integer overflow and underflow | Passed |
| Race conditions and reentrancy. Cross-function race conditions | Passed |
| Possible delays in data delivery | Passed |
| Oracle calls | Passed |
| Front running | Passed |
| DoS with Revert | Passed |
| DoS with block gas limit | Passed |
| Methods execution permissions | Passed |
| Economy model | Passed |
| Impact of the exchange rate on the logic | Passed |
| Malicious Event log | Passed |
| Scoping and declarations | Passed |
| Uninitialized storage pointers | Passed |
| Arithmetic accuracy | Passed |
| Cross-function race conditions | Passed |
| Safe Zeppelin module | Passed |
| Fallback function security | Passed |

# THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

## High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## Low Risk

Issues on this level are minor details and warning that can remain unfixed.

## Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

# FOUND THREATS

## ⚠ High Risk

Owner can change Tuna contract and add new token indexes.
This can be used to create artificial depositors with mock token which can result to contract's liquidity drain.

```
function addParams(address smartTuna, address smartPair) external onlyOwner
{ Tuna = smartTuna; Pair = smartPair; }

function addSupportedToken(uint tokenIndex, address addr, uint decimals) external onlyOwner
{
    require(tokenIndex > 0, "Dont change USDT token");
    SupportedTokens[tokenIndex].addr = addr;
    SupportedTokens[tokenIndex].decimals = decimals;
}

function ReplenishTuna(uint amount, uint tokenIndex) external nonReentrant
{
    require(_msgSender() == tx.origin, "Function can only be called by a user account");
    require(SupportedTokens[tokenIndex].addr == Tuna, "Parse token error");
    require(amount >= 1*10**18, "Min replenishment limit");
    _updatePercentage(_msgSender(), 0);
    _updateprePayment(_msgSender());
    user[_msgSender()].tuna += amount;
    tokenUSDT = IERC20(SupportedTokens[tokenIndex].addr);
    tokenUSDT.transferFrom(_msgSender(), address(this), amount);
}

function ReinvestTuna(uint amount) external nonReentrant
{
    require(_msgSender() == tx.origin, "Function can only be called by a user account");
    require(amount <= user[_msgSender()].tuna, "Insufficient funds");
    uint256 decimalsToken = 10**_decimals;
    uint256 tokenPrice = _getTokenPrice();
    uint256 swapExactly = ((amount * (tokenPrice * decimalsToken)) * 90 / 100);
    user[_msgSender()].tuna -= amount;
    user[_msgSender()].deposit += swapExactly;
    _updatePercentage(_msgSender(), 0);
    _updateprePayment(_msgSender());
}
```

# FOUND THREATS

## ⚠ High Risk

```
function _updatePercentage(address account, uint256 amount) internal
{
    uint tokenBalance = IERC20(Tuna).balanceOf(account) + amount;
    uint updPercentage;

    if(tokenBalance >= 100*(10**18) && tokenBalance < 200*(10**18))
        updPercentage = 120;
    else if(tokenBalance >= 200*(10**18) && tokenBalance < 500*(10**18))
        updPercentage = 140;
    else if(tokenBalance >= 500*(10**18))
        updPercentage = 160;
    else
        updPercentage = 100;

    if(user[account].percentage != updPercentage)
        user[account].percentage = updPercentage;
}

function _updateprePayment(address account) internal
{
    uint256 pending = pendingReward(account);
    user[account].timestamp = block.timestamp;

    if(pending > 0)
    {
        user[account].money += pending;
        user[account].earned += pending;
    }

    if(user[account].earned >= (user[account].deposit * 250 / 100))
    {
        user[account].deposit = 0;
        user[account].timestamp = 0;
    }

    // WorkDays counter
    uint256 newCounter = (block.timestamp - startTime) / 86400;
    if(newCounter > daysWork)
        daysWork++;
}

function pendingReward(address account) public view returns(uint256)
{
    uint256 RewardTime = (block.timestamp - user[account].timestamp) / 86400;
    RewardTime = (RewardTime >= 1) ? 1 : 0;
    return (((user[account].deposit / 100 * user[account].percentage) / 100) * RewardTime);
}
```

# FOUND THREATS

## ⚠ High Risk

```
function Withdraw(uint256 amount, uint tokenIndex) external nonReentrant
{
    require(_msgSender() == tx.origin, "Function can only be called by a user account");
    require(amount >= MIN_WITHDRAWAL, "Min withdrawal limit");
    require(tokenIndex == 0, "Parse token error");
    _updatePercentage(_msgSender(), 0);
    _updateprePayment(_msgSender());
    require(amount <= user[_msgSender()].money, "Insufficient funds");
    user[_msgSender()].money -= amount;
    user[_msgSender()].withdrawn += amount;
    tokenUSDT = IERC20(SupportedTokens[tokenIndex].addr);
    tokenUSDT.transfer(_msgSender(), amount);
}
```

# FOUND THREATS

## ⚠ High Risk

Fraudulent behavior scenario:
1. Contract's owner changes Tuna contract to mock Tuna token (newly deployed token without liquidity) via addParams() function.

2. Contract's owner creates new supported token index with address of the mock Tuna token via addSupportedToken() function.

3. Fraud account that have big supply (can be up to uint256's MAX value) of the newly added mock Tuna token uses the ReplenishTuna() function with big enough amount and token index that supports the mock Tuna token.

4. Fraud account that used the ReplenishTuna() function then uses the ReinvestTuna() function with big enough amount, via this function user's deposit and user's money are increased.

5. Fraud account uses withdraw() function and drains the available USDT liquidity amount in the contract.

# Informational

Daily Return Of Investment (ROI) starting from 1% up to 1.6% if the investor holds certain amount of Tuna tokens:
100 Tuna tokens: +0.2% daily ROI
200 Tuna tokens: +0.4% daily ROI
500 Tuna tokens: +0.6% daily ROI
*At time of this audit, current Tuna token address is: 0x75AdB3f6D788C344C409278263F70C5b60FeB33a

```
function _updatePercentage(address account, uint256 amount) internal
{
    uint tokenBalance = IERC20(Tuna).balanceOf(account) + amount;
    uint updPercentage;

    if(tokenBalance >= 100*(10**18) && tokenBalance < 200*(10**18))
        updPercentage = 120;
    else if(tokenBalance >= 200*(10**18) && tokenBalance < 500*(10**18))
        updPercentage = 140;
    else if(tokenBalance >= 500*(10**18))
        updPercentage = 160;
    else
        updPercentage = 100;

    if(user[account].percentage != updPercentage)
        user[account].percentage = updPercentage;
}
```

*Tuna contract is not in the scope of the current audit.*

RECOMMENDATIONS FOR

# GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

## Fishing Tuna

### GOOD PRACTICES FOUND

This is Return Of Investment (ROI) dapp.

ROI dapps can be very volatile.

TOKENOMICS

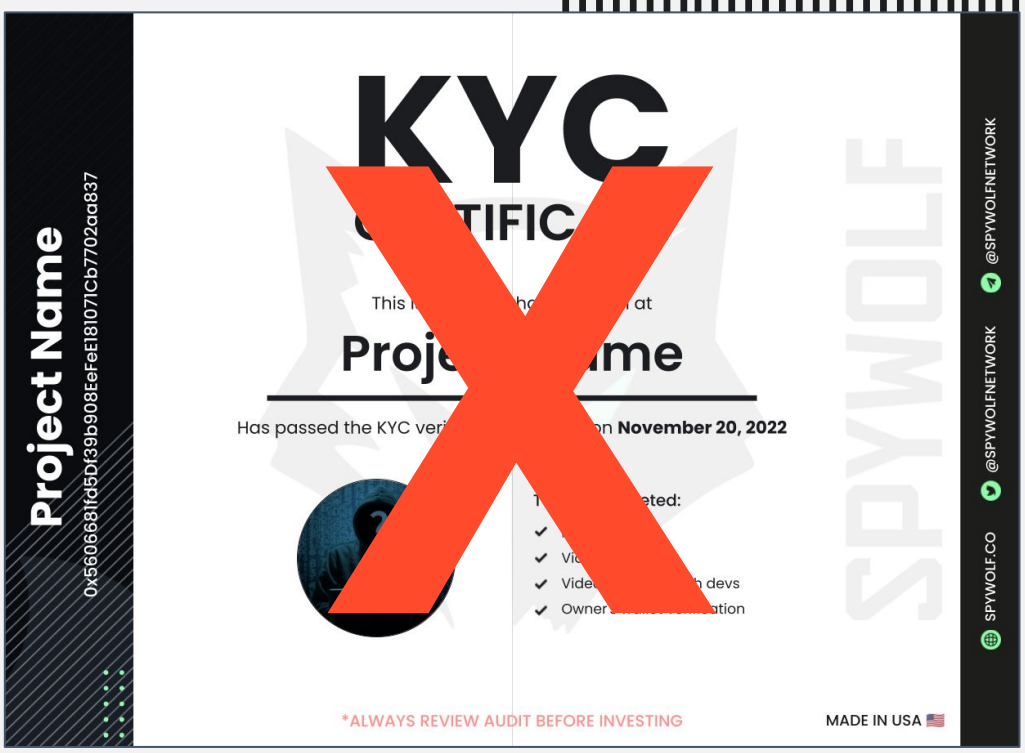
# THE TEAM

⚠️ The team is annonymous

## KYC INFORMATION

### No KYC

We recommend the team to get a KYC in order to ensure trust and transparency within the community.



WEBSITE

## Website URL
https://fishing-tuna.com/

## Domain Registry
https://joker.com

## Domain Expiration
2026-09-07

## Technical SEO Test
Passed

## Security Test
Passed. SSL certificate present

## Design
Very nice design with appropriate color scheme and overall layout.

## Content
The information helps new investors understand what the product does right away. No grammar mistakes found.

## Whitepaper
Well written, explanatory.

## Roadmap
No

## Mobile-friendly?
Yes

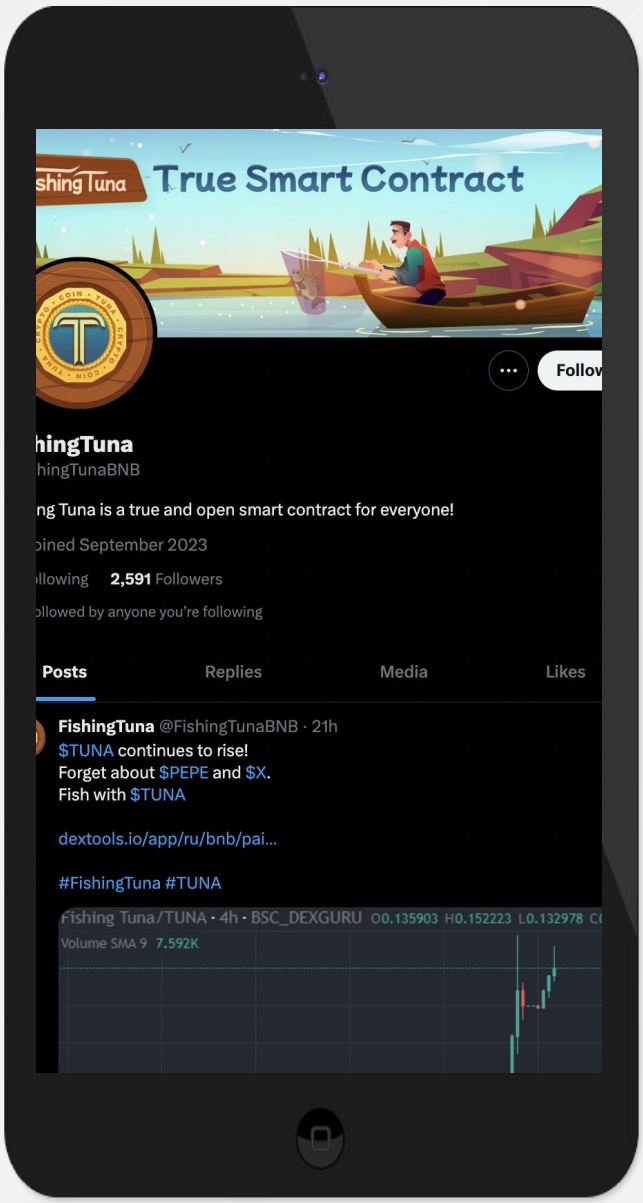

# fishing-tuna.com

# SOCIAL MEDIA

## & ONLINE PRESENCE

**ANALYSIS**

**Project's social media pages are active**




### Twitter

@FishingTunaBNB

- 2 508 followers
- Posts frequently
- Active

### Discord

- Not available

### Telegram

@fishing_tuna_chat

- 731 members
- Active members
- Active mods

### Medium

- Not available

# SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

# ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 500 SUCCESSFUL CLIENTS**
- **MORE THAN 500 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

## FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

# Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

**DISCLAIMER:**
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.